SYSTEM-ON-CHIP DESIGN FOR AUDIO PROCESSING

A dissertation submitted in partial fulfilment of the requirements for the degree of

MASTER OF TECHNOLOGY
IN
VLSI DESIGN AND EMBEDDED SYSTEMS

by

RAVI KANT BHUSHAN
ROLL NO: 213EC2194

To the

Department of Electronics and Communication Engineering
National Institute of Technology
Rourkela, Orissa, India
May 2015

SYSTEM-ON-CHIP DESIGN FOR AUDIO PROCESSING

A dissertation submitted in partial fulfilment of the requirements for the degree of

MASTER OF TECHNOLOGY
IN
VLSI DESIGN AND EMBEDDED SYSTEMS

by

RAVI KANT BHUSHAN
ROLL NO: 213EC2194

Under the Supervision of
Prof. A. K. SWAIN

to the

Department of Electronics and Communication Engineering
National Institute of Technology
Rourkela, Orissa, India
May 2015

DEPARTMENT OF ELECTRONICS AND
COMMUNICATION ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY, ROURKELA
ODISHA, INDIA-769008

CERTIFICATE

This is to certify that the thesis report entitled "**System-on-chip design for audio processing**" submitted by **RAVI KANT BHUSHAN**, bearing **roll no. 213EC2194** in partial fulfilment of the requirements for the award of **Master of Technology in Electronics and Communication Engineering** with specialization in "**VLSI Design and Embedded Systems**" during session 2013-2015 at National Institute of Technology, Rourkela is an authentic work carried out by her under my supervision and guidance.

To the best of my knowledge, the matter embodied in the thesis has not been submitted to any other university/institute for the award of any Degree or Diploma.

Place: Rourkela
Date: 1st June, 2015

Prof. A. K. SWAIN
Dept. of E.C.E
National Institute of Technology
Rourkela – 769008

Dedicated to
My beloved family

ACKNOWLEDGEMENT

With solemn respect and deepest gratitude, I would like to thank my project supervisor **Prof. A. K. SWAIN** who has always been the motivating force of this project work. His complete commitment to research work as well as tireless effort to gain knowledge and share it with his students had made him a true academician, who has become a source of inspiration for me. I am indebted to him for his valuable guidance, support throughout my project work as well as the good amount of time he had given to me to clarify my doubts and discuss about my work.

I express my sincere gratitude to Prof.(Dr) K.K.Mahapatra, Prof.(Dr.) P. K. Tiwari, Prof.(Dr.) D.P Acharya, Prof.(Dr.) Nurul Islam, and Prof. Shantanu Sarkar who had introduced the world of VLSI and Embedded System and helped me in grabbing knowledge in various domains of my specialization. I would also like to thank all other faculties and staff of ECE Department, NIT Rourkela for their help and support to complete my project work.

I am truly thankful to all research scholars of ECE Department, NIT Rourkela, especially Mr. George Tom, Mr. Sudeendra Kumar, Mr. Gopi Krishna S, Mr. VisweswaraRao, Mr. Jaganath Mohanty and Mr. Venkataratnam, who were always ready to share their knowledge throughout Our course. I also extend my gratitude to Mr. Anil Kumar Rajput, Mr. Nischay Malik and Mr. Ashutosh Singh for the worthy ideas we had shared on our respective research areas. I am really thankful to all my classmates and other friends who had made my stay in NIT a pleasant experience.

Lastly I thank my family whose constant support and encouragement, always help me move forward in life even during hard times.

Finally, I bow myself to Almighty God whose blessings guard and guide me throughout my life.

RAVI KANT BHUSHAN

ABSTRACT

Nowadays System-on-Chip (SoC) is present in every electronic system. SoC popularity is based on higher performance, reduced size, less power consumption, and alleviation of time to market by design reuse. Device scaling enabled SoC to integrate more functionality into a single chip and hence system complexity, like Audio Processing system, is no more barriers for the SoC designer. Speaker recognition/verification is one of the applications in biometrics for preventing identity fraud. It is suitable for real time scenarios and remote recognition over phone. In this project, I have designed a SoC system for Audio Processing on Altera DE2 board, FPGA platform, which automatically verify or recognize the speaker Identity. Mel Frequency Capestral Coefficient (MFCC) is used for feature extraction of the voice signal. Large samples of extracted feature are used to train the system by using Backpropagation Neural Network. After training, speaker verification done in real time by first extracting speaker voice feature, applying trained network on extracted feature, and comparing it with the stored database. Experimental result shows that the designed system is able to verify person's identity.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

SoC	System-on-Chip
SoPC	System-on-Programmable Chip
IP	Intellectual Property
RTL	Register Transistor Logic
FPGA	Field Programmable Gate Array
VHDL	VHSIC Hardware Description Language
MFCC	Mel Frequency Cepstral Coefficient
FFT	Fast Fourier Transform
DCT	Discrete Cosine Transform
nntool	Neural Network Tool
MSE	Mean Squared Error
LED	Light Emitting Diode
LCD	Liquid Crystal Display
RAM	Random Access Memory
DRAM	Dynamic Random Access Memory
PLL	Phase Locked Loop
PIO	Parallel Input/Output
CODEC	Compression/Decomprassion
JTAG	Joint Test Action Group
UART	Universal Asynchronous Receiver-Transmitter
EDS	Embedded Design Suit

Chapter 1

Introduction

The fields like mobile communication, digital signal processing gained rapid growth and provoked the design engineer to develop complex systems into a single chip i.e. System-on-Chip (SoC). Core of every electronic system in today's life, mobile phone to spacecraft, remote controlled toy car to missile control, is a SoC. SoC design methodology combines IP cores of embedded processors, memory blocks, interface blocks, and analog blocks on a single chip [1]. It is found that this single chip system has good performance, reduced size, and less power consumption than conventional design methodology.

Speaker Recognition System, an Audio Processing, evolution started in early 1960's [2,3] with voiceprint analysis, where uniqueness of an individual is characterized by the characteristics of an individual voice. The detection efficiency of Speaker Recognition systems gets severely affected in noise presence. This reality ensured to derive a more reliable method. In Speaker Recognition process, acknowledgement of the speaker is based on some characteristics match of the speech wave with the stored database. In general three phases are required in Speaker Recognition systems (Figure 1.1). The first phase is Acoustic Processing where sampling of voice is done with start and end detection module to process only voice signal. The second phase is Feature Extraction where the exceptional features of the speaker voice are extracted. In last phase, Feature Matching is done where speaker extracted voice features is compared with the catalog of acknowledged speakers stored in memory. The efficiency of the Speaker Recognition System depends on efficient feature extraction and comparison algorithm between real time voice sample and stored database.

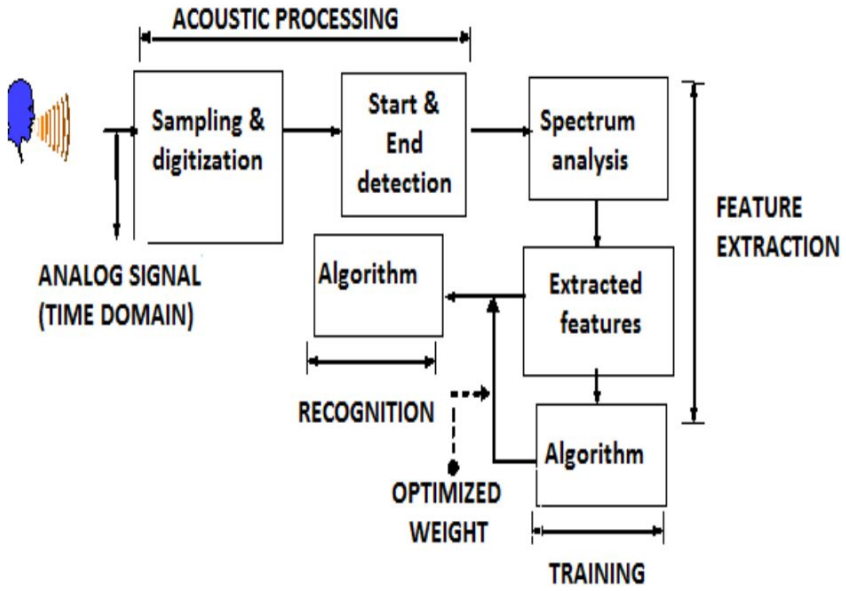


Figure 1.1: Basic structure of Speaker Recognition System

1.1 Application of Speaker Recognition Systems

- For security application: password protected lock system for our home, locker, computer etc.
- For crime investigations: verifying voice of criminal from the audio tape of telephonic conversations.
- For alternate verification of personal identification number, credit card number, and telephone number.

1.2 Motivation

SoC design methodology combines necessary hardware and electronic circuitry like embedded processors, memory blocks, interface blocks, and analog blocks on a single chip. Technology advancement in semiconductor design made much functional integration in SoC. Single chip integration leads to good performance, reduced size, and

less power consumption than conventional design methodology. SoC enables designer to add or change a block to accommodate late alterations, reuse of microcontroller or discrete peripherals IP cores.

Fraudulent multiple identities or identity fraud is a serious problem in areas like financial transaction, law enforcement, network management security, etc. Biometrics is a promising solution in market security and personal verification in which speaker recognition system is one of them. Research challenges in speaker recognition are speaker voice change due to aging, illness, and emotions, amount of speech in training, and mismatch in voice recording conditions between training and testing.

1.3 Literature Survey

Research in Speaker Recognition System has been focused on Voice Feature. Feature of voice is then used to develop a reliable, robust and efficient Recognition System. However, voice feature is highly affected due to individual speaker characteristics, emotion variations and noise disturbance. These variations increases system complexity and hence provide challenges in robust system designing.

Template-matching techniques are being used for Text-dependent methods. Spectral Feature Vector of the input voice signal used for speaker recognition. To aligning input speech in time axis and each template of the registered user is done by using Dynamic Time Warping (DTW) Algorithm [4]. Accumulation from beginning to the end of the speech, the degree of similarity between the calculated Statistical variations in spectral features can be modeled by Hidden Markov Model (HMM).

HMM-based method is expansion of DTW-based method. A new technique for computing verification scores using multiple verification feature from the list of scores for the target speaker's speech was introduced [5]. This technique was compared to the baseline logarithmic likelihood ratio verification score using global Gaussian Mixture Model (GMM) speaker models. It gave no improvement in verification performance.

Neural and Fuzzy technique is used by Gupta, CHEEDELLA S. [6]. He has applied the technique for Speaker Independent Speech Recognition System. He tested vide numbers of speech templates from the persons belonging to different areas and in noisy environment. The resultant system gave 92.2% recognition rate.

Zhonghua, Fu, and Zhao Rongchun [7] proposed a speaker verification system which used group of Neural Networks instead conventional single network for pattern recognition. For pattern recognition Supervised Learning Vector Quantization (LVQ) has been used. For the system having bigger number of speakers, recognition rate gets severally affected. To overcome this decreased recognition rate they come up a new solution of hybrid feature parameter vector which is developed by Linear Pridictive Coding (LPC) and Cepstral Signal Processing technique.

Saha and Yadhunandan [8] has proposed modified Mel-Frequency Capstral Coefficien (MFCC) feature. Discriminative ability is compared for performance measure by using Multi-Dimensional F-ratio. The same performance could be also got by using Bark scale (Aronowitz et al, 2005). Revised Perceptual Linear Prediction Coefficient (RPLP) is proposed by Kumar et al, (2010), Ming et al, (2007) which is combination of MFCC and PLP.

1.4 Organization of Thesis

- Chapter 2 introduces System-on-Chip, Altera DE2 Development and Education Board (FPGA platform for SoC design), and Software, a GUI for Hardware Description Language and C language for designing the system.
- Chapter 3 describes speaker recognition principle. Mel-Frequency Cepstral Coefficient (MFCC) is used for voice feature extraction. Backpropagation Neural Network is used for training and developing the network. Feature matching for speaker verification in real time.
- Chapter 4 describes implemented SoC on Altera DE2 board. Hardware resources and software flow.
- Chapter 5 shows results of hardware resources used on Altera DE2 board, training neural network in term of performance, and real time system testing.
- Chapter 6 concludes the work done with an insight into future work.

Chapter 2

System-on-Chip

A System-on-Chip (SoC or SOC) is a complete system into a single semiconductor chip with necessary hardware and electronic circuitry. Cellular phone,

digital camera, set-top box, PDAs, etc are System-on-Chip. It also includes application in nanotechnology and medical technology.

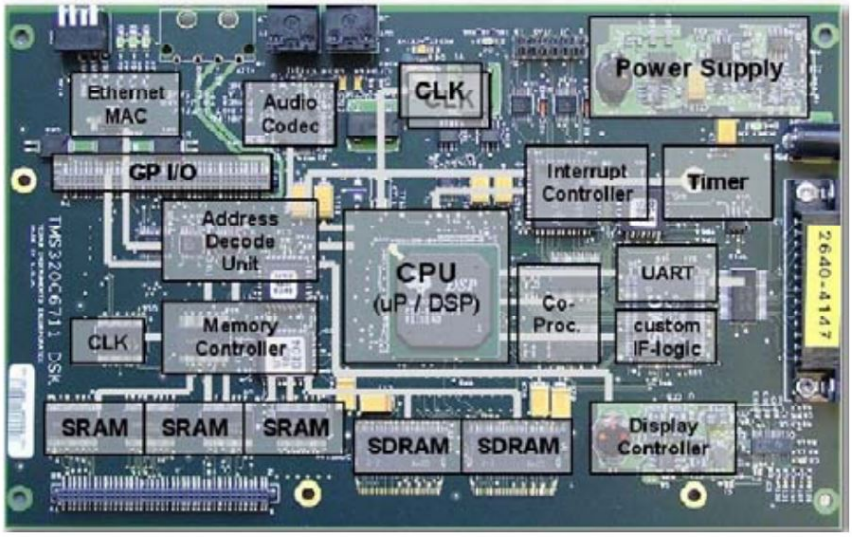


Figure 2.1 Example of SoC

Basic SoC elements are:

- Microprocessor, Microcontroller, Digital Signal Processing (DSP) core.
- Memory Blocks: RAM, ROM, EEPROM, Flash
- Timing source: Oscillator and Phase-Locked Loop (PLL).
- Peripherals: Counter, Timer, Power-On Reset generator etc..
- External Interface: Ethernet, USB, SPI, USART etc.
- Analog Interface: ADC and DAC.
- Voltage Regulators and Power Management Circuits

Advantages:

- Consumes less power, low cost, higher reliability.

- Creating custom chips in hours rather than months.
- Add or change a block to accommodate late alterations.
- Reuse of μC or discrete peripherals IP cores.
- Several IP blocks availability benefits designer code software program to control and communicate the IP blocks.

Fabrication technology for SoCs:

- *Full Custom:* Each individual transistor layout and interconnects are designed.
- *Standard Cell:* Some standard functionality component is available which is known as standard cell.
- *Field Programmable Gate Array (FPGA):* logic gates and programmable interconnects are already fabricated on single chip. Desired functionality system is achieved by simply custom hardware programming.

2.1 Altera DE2 (Development and Education Board)Board

Altera DE2 Board is ideal platform for designing System-on-Chip in the multimedia, storage, and networking field. It is FPGA kit whose custom hardware programming is done on Quartus II CAD tool. It has multiple features which make the board appropriate for laboratory use in university and college courses, for multiple projects, in addition to for designing refined digital systems.

Figure 2.2 shows layout of the Alera DE2 board with position of the components and connectors.

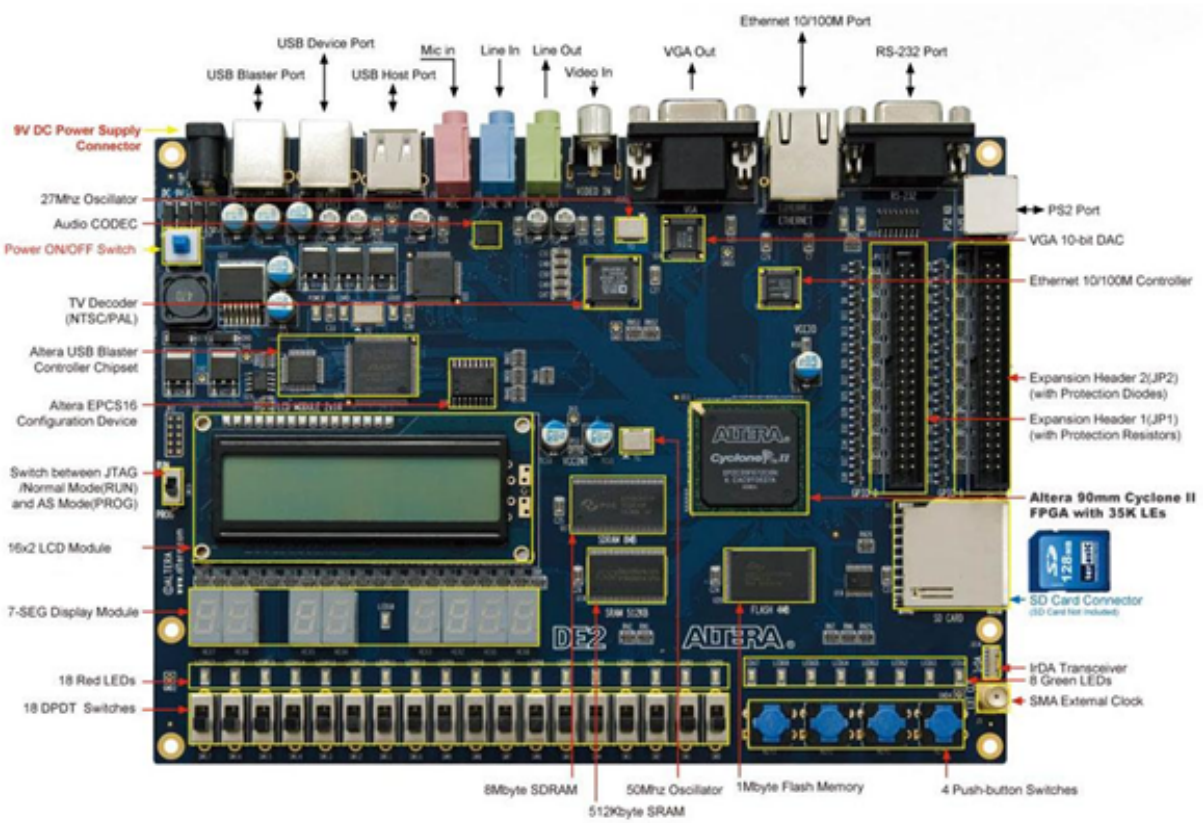


Figure 2.2 The Altera DE2 board.

2.1.1 Altera DE2 board Hardware:

- Altera Cyclone® II 2C35 FPGA device
- Altera Serial Configuration device - EPCS16
- USB Blaster (on board) for programming and user API control; both JTAG and Active Serial(AS) programming modes are supported
- 512-Kbyte SRAM
- 8-Mbyte SDRAM
- 4-Mbyte Flash memory
- SD Card socket
- 4 pushbutton switches

- 18 red user LEDs
- 9 green user LEDs
- 50-MHz oscillator and 27-MHz oscillator for clock sources
- 24-bit CD-quality audio CODEC with line-in, line-out, and microphone-in jacks
- VGA DAC (10-bit high-speed triple DACs) with VGA-out connector
- TV Decoder (NTSC/PAL) and TV-in connector
- 10/100 Ethernet Controller with a connector
- USB Host/Slave Controller with USB type A and type B connectors
- RS-232 transceiver and 9-pin connector
- PS/2 mouse/keyboard connector
- IrDA transceiver
- Two 40-pin Expansion Headers with diode protection

2.1.2 Altera DE2 Board Block Diagram

All peripherals are connected to Cyclone II FPGA device. It gives flexibility to user to use any number of peripherals for the system design.

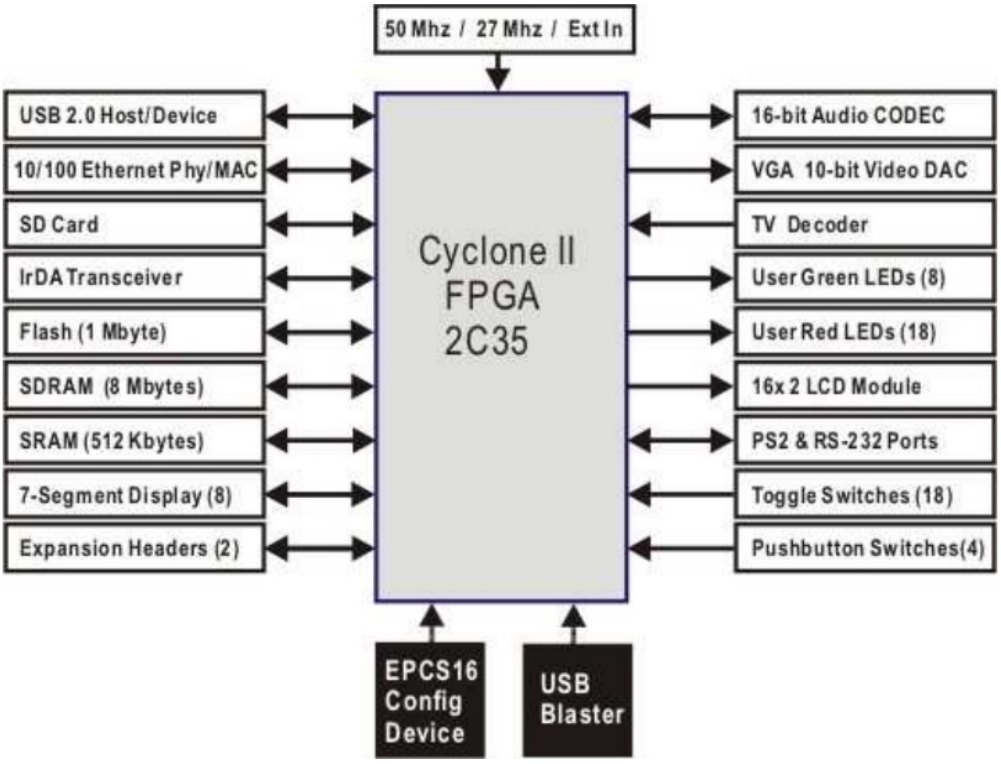


Figure 2.3 Block diagram of the DE2 board

Properties of Cyclone II 2C35 FPGA

- 33,216 LEs
- 105 M4K RAM blocks
- 483,840 total RAM bits
- 35 embedded multipliers
- 4 PLLs
- 475 user I/O pins
- Fine Line BGA 672-pin package

2.1.3 Nios II Processor

Altera DE2 board FPGA contains configurable 32 bit Embedded Processor named as NIOS II. NIOS II is more appropriate for Embedded Computing Applications, like DSP, Control System.

NIOS II Processor Architecture:

- Classic Pipelined RISC Machine
- 32 General Purpose Registers
- 3 Instruction Formats
- 32-Bit Instructions
- 32-Bit Data Path
- Flat Register File
- Separate Instruction and Data Cache (configurable sizes)
- Tightly-Coupled Memory Options
- Branch Prediction
- 32 Prioritized Interrupts
- On-Chip Hardware (Multiply, Shift, Rotate)
- Memory Management Unit (MMU)
- Memory Protection Unit (MPU)
- Custom Instructions
- JTAG-Based Hardware Debug Unit

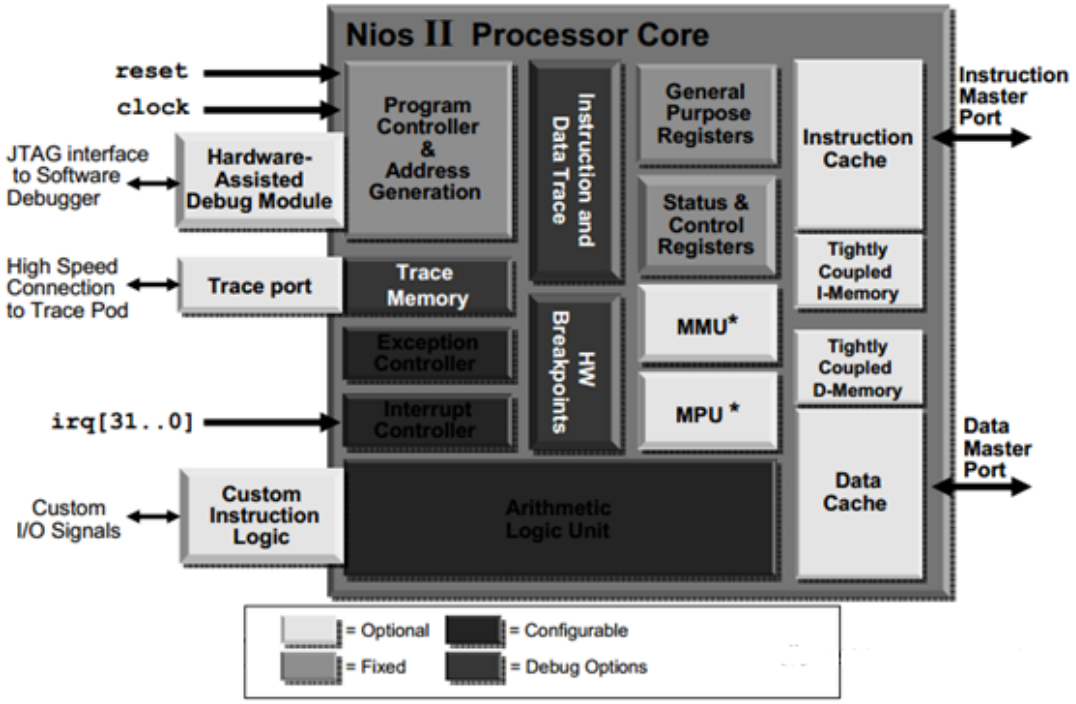


Figure 2.4 Block diagram of Niso II Processor.

2.1.4 Software Package for Altera DE2 Board

1. **Quartus II** → The Quartus II Development Software is used for designing programmable logic device. It incorporates an implementation of VHDL and Verilog for Hardware description, visual editing of logic circuits, and vector waveform. Hardware developer can compile his/her design, inspect RTL diagram, execute timing analysis, examine design response on various stimuli, and configure the target device with the programmer.
2. **Qsys (SOPCBuider)** → The Qsys System Integration Tool automatically generates interconnects logic to interface Intellectual Property (IP) function and subsystem. It integrates a library of ready-made components and also provide

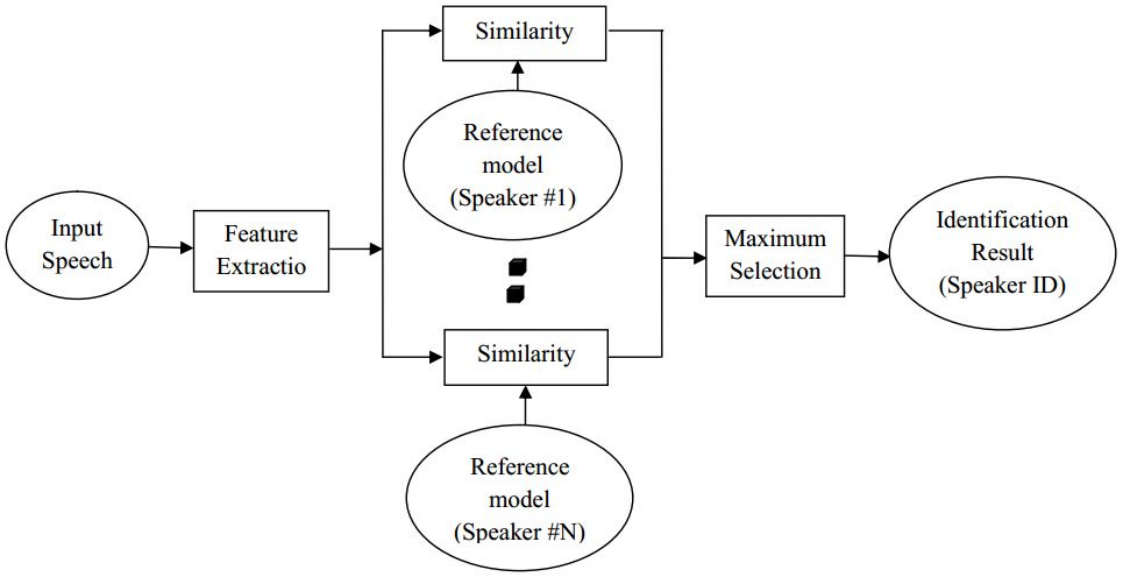
interface to incorporate custom component. While generating the system, Qsys automatically handles bus arbitration, bus width, and clock domain crossing.

3. **Nios EDS (Embedded Design Suit)** → The Nios II Embedded Design Suite (EDS) provides platform to design software for NIOS II processor. It contains device drivers, hardware abstraction layer (HAL) library, commercial grade network stack etc. for the designated NIOS II processor. It can also evaluate a real time operating system.

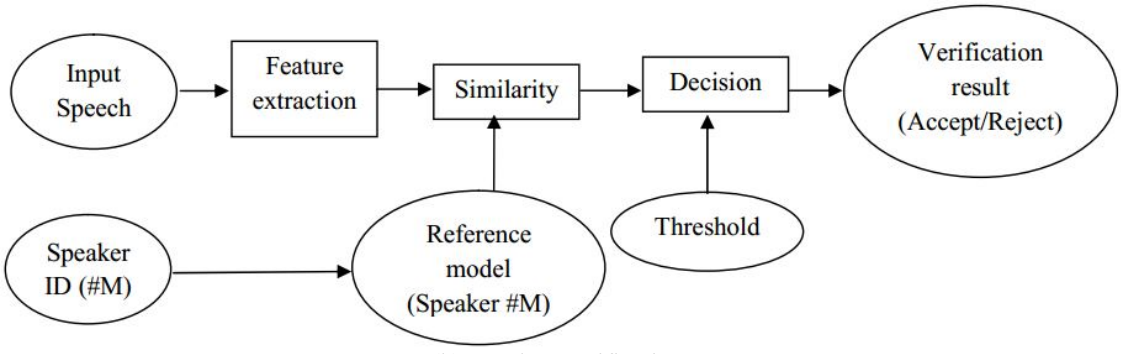
Chapter 3

Principle of Speaker Recognition

Speaker Identification and *Speaker Verification* are the two modes of Speaker Recognition System. In *Speaker Identification* mode, the system detects the speaker which are already registered in the system by processing his/her voice. In *Speaker Verification* mode, the system accepts or rejects the identity claim of the speaker. Basic structure of *Speaker Identification* and *Speaker Verification* is shown in the Figure 3.1. My project is to design a *Speaker Identification System*, so only Speaker Verification is described in the next section.



(a) Speaker Identification



(b) Speaker Verification

Figure 3.1. Basic structures of Speaker Recognition Systems

All Speaker Recognition System has two main modules: *Feature Matching* and *Feature Matching*. In *Feature Extraction,* the exceptional features from the speaker voice are extracted which is used to represent the speaker. In *Feature Matching(testing),* extracted feature of speaker voice is compared with database of stored set of known speakers and provides regarded output. So for Feature Matching, first system needs to be trained and generate reference model by taking voice sample from speaker to be registered.

3.1 Voice Feature Extraction

Speech signal is quasi stationary i.e. it varies slowly with respect to time. But when it is examined over adequately short period of time (less than 100 msec), it seems stationary. *Short Spectral Analysis* for characterizing voice is possible for that period of time. Voice signal is shown in the figure 3.2.

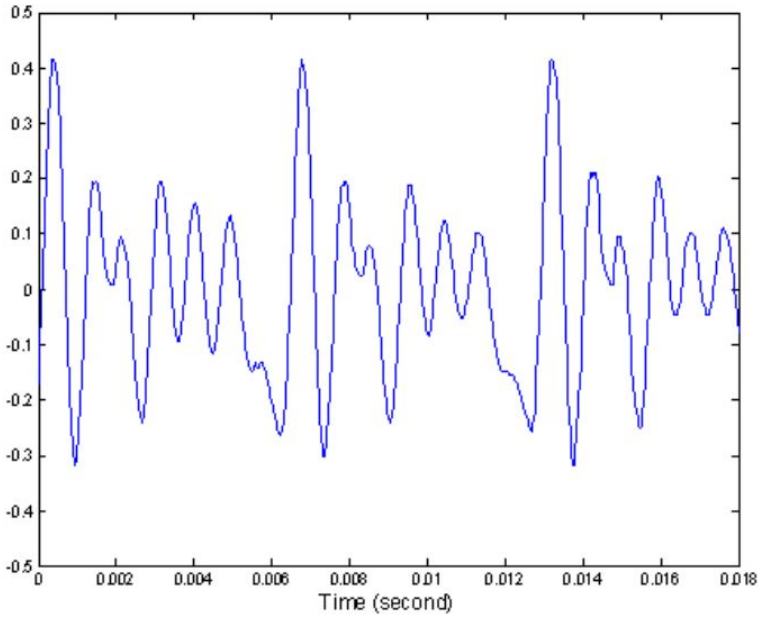


Figure 3.2 Example of voice signal

Liner Predictive Coding (LPC), Mel Frequency Capstrum Coefficient (MFCC) etc. are some of the known extracted Voice Feature. I am using MFCC for my project. MFCC mimics variation of human ear's critical bandwidth with frequency i.e. varies linearly in low frequency (below 1000 Hz) and logarithmic in high frequency (above 1000 Hz).

3.2 Mel-Frequency Cepstrum Coefficients processor

Figure 3.3 shows block diagram of MFCC processor. Human voice frequency lies below 4 KHz, so for the system to process the speaker voice must record the voice signal with sampling frequency more than 8 KHz to avoid aliasing effect. For voice signal sampling an Audio CODEC is required, and the output sampled data is further processed by the function to get MFCC which is given in the Figure 3.3. Each function block in the given figure is described briefly in the following topic.

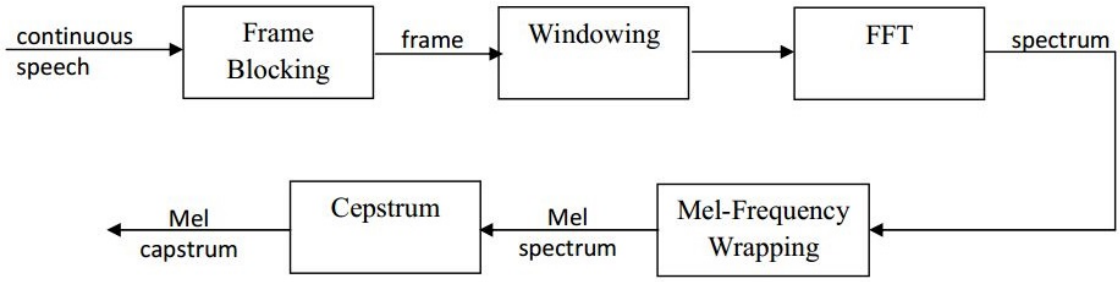


Figure 3.3 Block diagram of the MFCC processor

3.2.1 Frame Blocking

In Frame Blocking, the sampled voice signal is blocked in frames of N samples. Each frame samples are separated by M (M<N) samples from the adjacent frame sample

and overlap the previous sample with N-M samples. If the sampling frequency is 8 KHz then the typical values of N=256 (~30 msec voice duration) and M=100.

3.2.2 Windowing

The main purpose of Windowing the input frame of samples is to minimize the voice signal discontinuity which might be introduced in Frame Blocking. This helps in minimizing the spectral distortion. Windowing function taper the frame samples at the beginning and end of the frame. If window of frame containing N samples is defined as, w(n)=0≤n≤N-1, then Windowed single is given by

$$y_1(n) = x_i(n)w(n), \quad 0 \le n \le N-1$$

In this project, I am using Hamming Window whose window function is defined as:

$$w(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right), \quad 0 \le n \le N-1$$

3.2.3 Fast Fourier Transform (FFT)

Purpose of the Fast Fourier Transform (FFT) is to convert time domain signal into frequency domain signal. It is the fast algorithm for Discrete Fourier Transform (DFT) which can be implemented on digital hardware. In MFCC processor, it applies on each frame block. FFT on N samples with samples $\{x_n\}$ is defined as:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-j2\pi kn/N}, \qquad k = 0,1,2,\dots\dots,N-1$$

FFT output (X_k) generally gives a complex number. For my system, I calculated its absolute value. As FFT is double sided i.e. it contains both positive and negative

frequency so I accept only positive frequency output and discarded negative frequency output.

3.2.4 Mel-frequency Wrapping

According to psychophysical studies, it has been determined that the human perception for the frequency content in voice is not linear. To mimic human ear perception a filterbank with liner below 1000 Hz and logarithmic above 1000 Hz is designed by 'mel' and is defined as:

$$m = 2595 \log\left(\frac{f}{700} + 1\right)$$

Mel filterbank provides and output corresponding to voice tone fundamental frequency. The mel filterbank is shown in figure 3.4. It has triangular bandpass frequency response with bandwidth and spacing related to mel frequency interval which is chosen according to desired spectrum coefficient and frequency range.

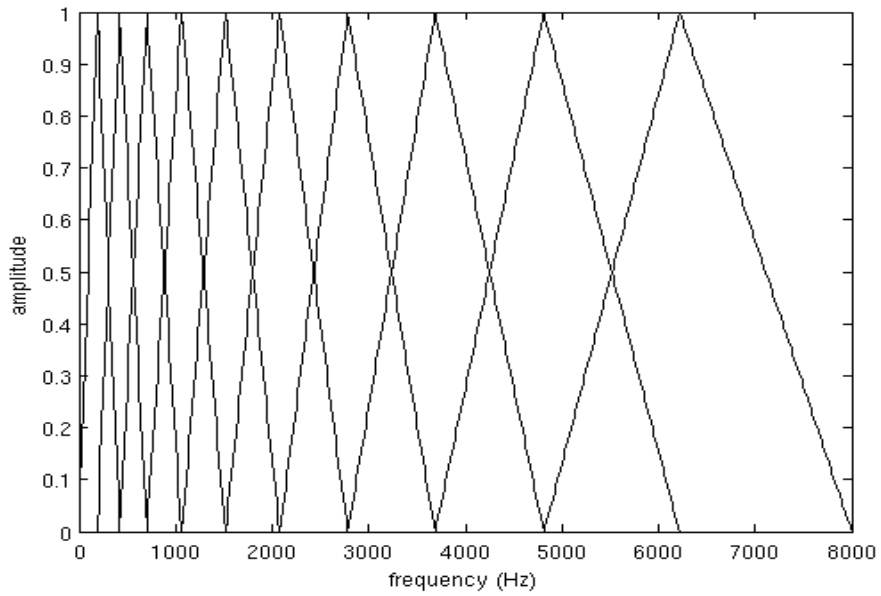


Figure 3.4 Example of Mel Filterbank

3.2.5 Cepstrum

In this step in MFCC processor we convert log mel spectrum back into time domain. Discrete Cosine Transform (DCT) is used for this purpose. This is the final output that contain voice local spectral feature. For the given mel log spectrum coefficient $\widetilde{S},\ k = 0,2, \dots \dots k-1,$ the time domain coefficient (MFCC) $\widetilde{c_n}$ is calculated as:

$$\widetilde{c_n} = \sum_{k-1}^{k} (\log \widetilde{S_k}) \cos\left[n\left(k - \frac{1}{2}\right)\frac{\pi}{K}\right], \qquad n = 0,1,2, \dots, K-1$$

As first segment represent mean estimation of the voice signal which has less information about voice signal, it can be excluded.

3.3 Feature Matching

Feature matching involves assigning speech signals of each speaker a different class based on its feature. Features are taken from known samples and then unknown samples are compared with those known samples. Different techniques such as Neural Networks, Minimum distance classifier, Bayesian classifier, Quadratic classifier, Correlation are used for this purpose. In this project, I have opted for Artificial Neural Networks.

In my project, I am designing a system which runs in real time, so it is required to choose simplest neural system which processes fast and has good efficiency. For this purpose I am using Multilayer Feedforward Neural Network. In this network I am having input, Hidden layer, and predicted output. Figure 3.5 shows a 2 layer feedforward neural network with 12 input nodes, 5 hidden nodes, and single output node.

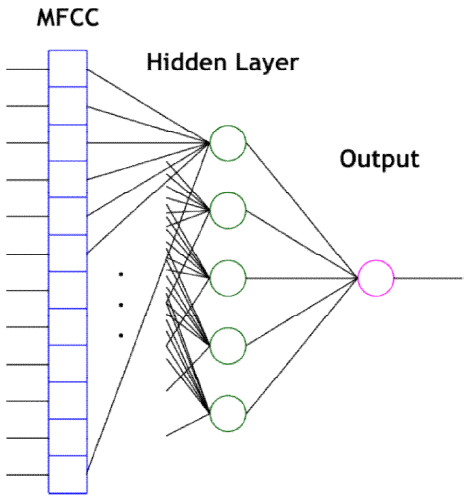


Figure 3.5 Two layer feedforwared neural network

Figure 3.6 shows hidden node calculation in feedforwared neural network in which each node of hidden layer is calculated by summing of weighted inputs. Some offset bias may also be summed up. The same calculation is also done for output where weighted hidden nodes are summed up with bias.

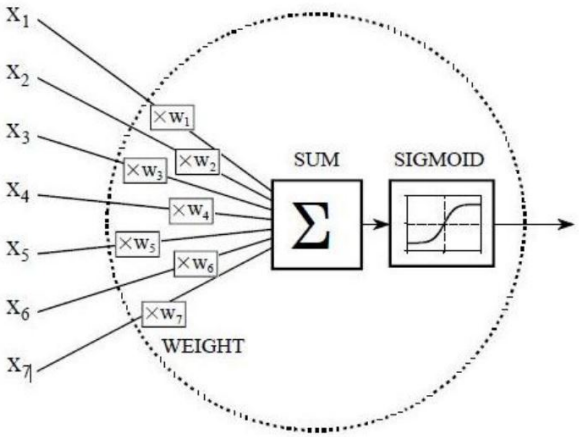


Figure 3.6 Node calculation in Neural Network

Individual hidden nodes and output of Multilayer Feedforwared Neural Network is:

$$f(x) = \tanh\left(b + \sum w_i x_i\right)$$

In the above equation, x is the input vector or hidden vector (in output calculation), b is the offset bias, w is the weight vector. To tame the result hyperbolic tangent is used so as to get hidden node and output value in the range -1 to 1.

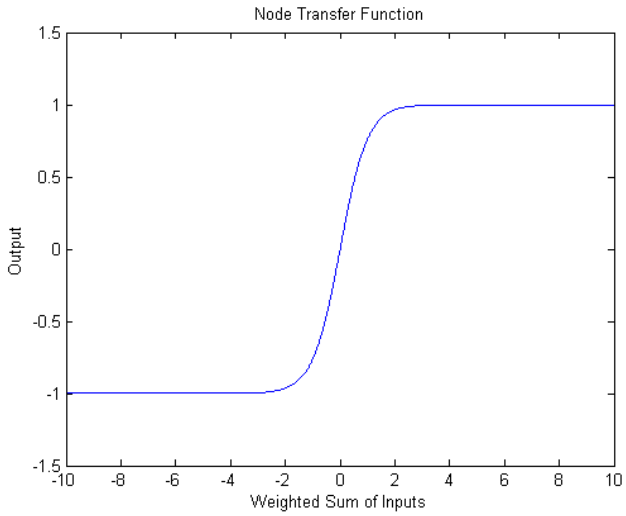


Figure 3.7 Activation Function: Hyperbolic Tangent

3.4 Training

In Feature Matching, extracted voice features are compared with the database of the registered speakers which is stored in memory. So the system first needs to develop database. For this purpose system goes through training phase. For training purpose I am using *Backpropagation Neural Network*. In training phase, enough sample of voice signal is provided to the system by the speaker to be registered. By using these samples and Backpropagation Neural Network algorithm, system develop database by approximating

non-linear relationship between input and output by modifying the values of weight internally.

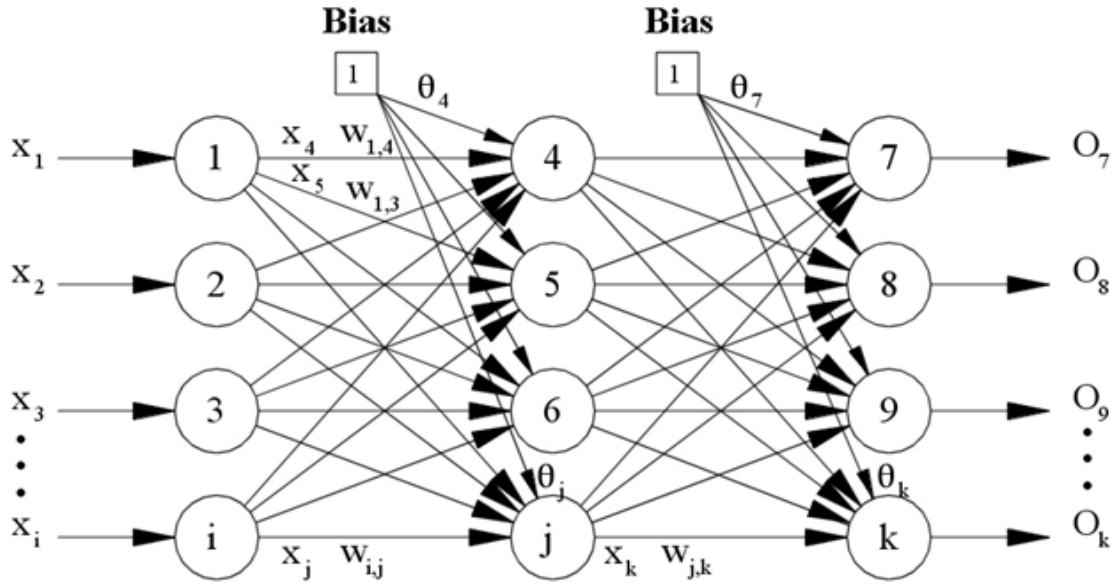


Figure 3.8 Backpropagation Neural Network with one hidden layer

There are two steps for Backpropagation Neural Network operation: *feedforward* and *backpropagation*. In the feedforward step, an output is calculated by applying weight and bias to the input layer according to the model used and the same for other layer respectively. Then the calculated output and target output is being compared to get the error signal. This error signal to the output is the contribution by all hidden nodes, so this output error is transmitted backward to the adjacent hidden nodes with output layer. This process follows layer by layer in the network until each node in the network receives the error signal that describes the relative contribution to the overall error. These each node error is used to updated weight and bias for each layer, then whole steps is repeated again and again until the error function in weight space has satisfactory delta rule of gradient

descent. The weight that has minimum error function is considered to be solution for the learning problem.

Algorithm

Figure 3.8 shows Neural Network with i input nodes ($X_1, X_2, X_3, \ldots\ldots, X_i$), one hidden layer with j nodes, and k output nodes ($O_1, O_2, O_3, \ldots\ldots, O_k$), weight w, and bias θ. We will use Levenberg-Marquardt Backpropagation Algorith.

Feedforward step:

In hidden layer, j^{th} node is given by-

$$\text{Net}_j = \sum w_{i,j}\, x_j + \theta_j \qquad (1)$$

Net_j is the weighted sum of the input neuron. θ_j is the offset bias which is used to make non zero node that makes the neural network to be trained.

Next step is to pass Net to a suitable Activation Function which is a bounded differentiable real function, defined for all real input value, and has positive derivative at each point. It gives output of the neuron which becomes input for the next layer of the network. I have used Hyperbolic Tangent Sigmoid Transfer Function.

$$O_j = X_k = \tanh(Net_j) \qquad (2)$$

Same steps is used to get output nodes using hidden layer nodes as an input node.

Backpropagation (Error Calculations and Weight Adjustments):

Let O_k is the calculated activation output node k, and t_k is the expected output node k, then difference between them is given by:

$$\Delta_k = t_k - O_k \tag{3}$$

The error signal of k^{th} node output is:

$$\delta_k = \Delta_k O_k (1 - O_k)$$

or

$$\delta_k = (t_k - O_k) O_k (1 - O_k) \tag{4}$$

The term $O_k(1-O_k)$ is derivative of the Sigmoid Function. So weight of node connection j to k is proportional to the error at node k multiplied by the activation of j node.

Modified weight W_{jk} between node j and k is:

$$\Delta w_{j,k} = l_r \delta_k x_k \tag{5}$$

$$w_{j,k} = w_{j,k} + \Delta w_{j,k} \tag{6}$$

In the above equation, ΔW_{jk} is the weight change between nodes k and j, l_r is learning rate.

Network learning performance depends upon l_r i.e. if it is too low, learning rate will be slow, and if it is too high, oscillation around minimum point occur and will weight adjustment will never reached. For the latter case some modification in Backpropagation Algorithm helps the learning rate to reduce from large value and this leads to reach to the optimal point of minima. So the modified equation (5) for updating weight is given as:

$$\Delta w_{j,k}^{n} = l_r \delta_k x_k + \Delta w_{j,k}^{(n-1)} \mu \quad (7)$$

In the above equation, a momentum term (μ) is included during n^{th} iteration, which is multiplied to the $n\text{-}1^{th}$ iteration of the W_{jk}. The momentum term accelerate the learning process, hence weight change. Generally momentum term value lies between 0 and 1.

Hidden Layer:

Error signal for the hidden layer node j is evaluated as

$$\delta_k = (t_k - O_k) O_k \sum(w_{j,k} \delta_k) \quad (8)$$

In the above equation, for the output layer, weighted error signal of all nodes k is summed up.

Equation for adjusting weight W_{ij} between node I and j is

$$\Delta w_{i,j}^{n} = l_r \delta_j x_j + \Delta w_{i,j}^{(n-1)} \mu \quad (9)$$

$$w_{i,j} = w_{i,j} + \Delta w_{i,j} \quad (10)$$

Global Error

For minimizing output error, the following error function is developed for all pattern

$$E = \tfrac{1}{2} \sum (\sum (t_k - O_k)^2) \quad (11)$$

Zero error function is an ideal case but practically it is not possible. So lowest value should be appreciated.

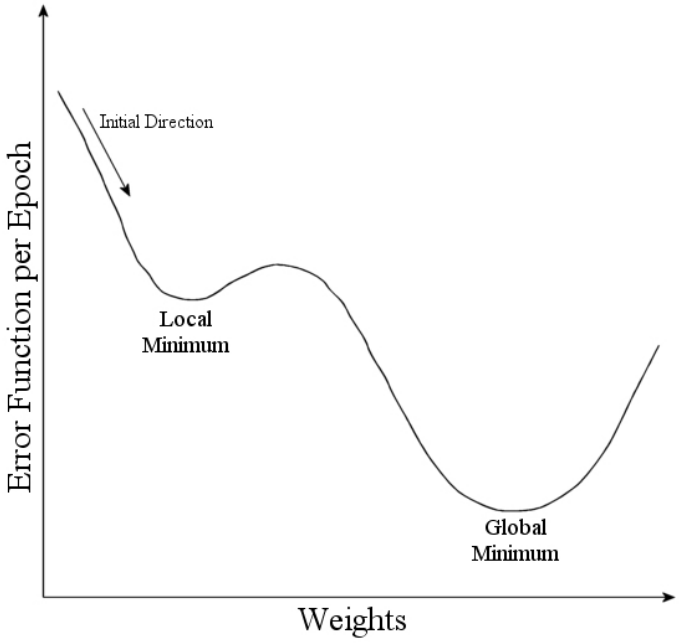


Figure 3.9 Global and Local Error

Chapter 4

SoC Hardware and Software Implementation

Full system design contains Hardware and Software design. For Hardware design, I am using Altera DE2 FPGA board and Nios II EDS for Software.

4.1Hardware Design

Hardware for SoC Audio Processor is designed on Altera DE2 Board which has Cyclone II FPGA (Figure 4.1). I have used Verilog HDL language to describe my Hardware and after confirming my design I have dumped it on the FPGA. The top level module contains NIOS II processor, Memory module, FFT controller, Audio module with I2C bus controller to configure Audio CODEC. For clock requirement for SDRAM, NIOS II processor, FFT, Audio CODEC two PLLs is used

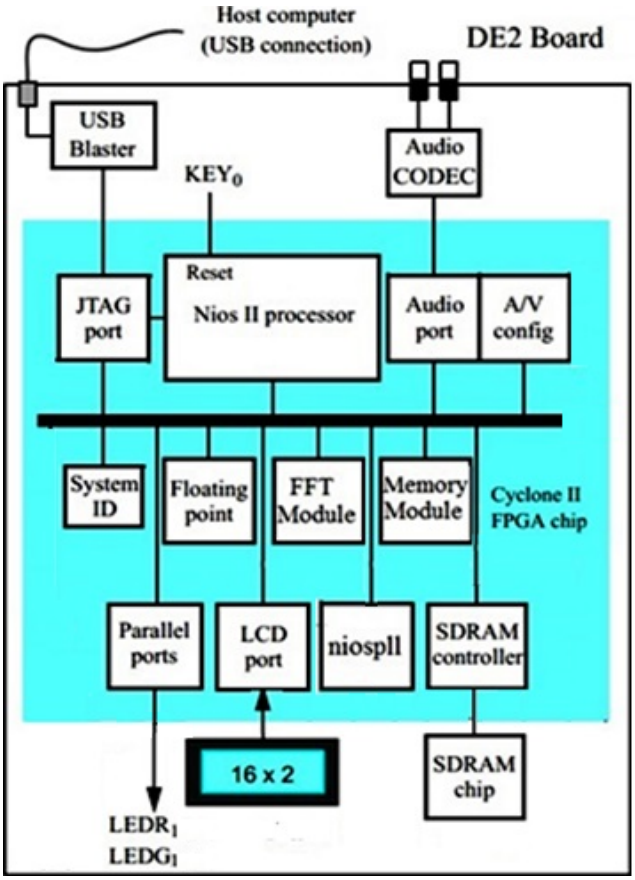


Figure 4.1 Block Diagram of Audio Processor

SDRAM

- Used for storing the NIOS II program and for memory requirement during NIOS II Processor runtime.
- 8-Mbyte Single Data Rate Synchronous Dynamic RAM memory chip
- Organized as 1M x 16 bits x 4 banks

Clock inputs

- 50 MHz, for NIOS II Processor and SDRAM
- 27 MHz for Audio CODEC

Audio CODEC

- Used for capturing Audio signal from speaker
- Wolfson WM8731 16-bit sigma-delta audio CODEC
- Microphone input jacks
- Sampling frequency: 32 KHz

JTAG UART Port

- Used for Serial Character Stream communication between Host PC and SOPC Builder, and for Debugging.
- Debugging Level → 2
- It download
    - software
    - software breakpoints
    - 2 hardware breakpoints
    - 2 data triggers

16x2 LCD Display

- Used to display result
- Model : CFAH1602B-TMC-JP
- Company : Crystalfontz America, Inc.
- 16 character x 2 Line
- Dimension → 80.0 x 36.0 x 13.5(MAX)mm
- Character size →(L)2.95 x (W)5.55mm
- LCD type →STN, Negative , transmissive, Blue
- Backlight →LED White

Pushbutton switches

- Input for Reset
- 2 nos. Pushbutton Switches
- Schmitt Trigger Circuit debounce
- Normally high; generates one active-low pulse when the switch is pressed

LEDs

- Used for indicator to show the speaker verified or not
- 1 RED LEDs
- 5 GREEN LEDs
- Cyclone II FPGA controlled

Memory Module

- Used to store a block sequence of samples received from Audio CODEC
- RAM megafunction is used to block required size of memory.
- Memory width → 16 bits

- Memory Size → 1024 x 16 bits

I2C_AV_Config (I2C AV Configuration)

- The Audio Video Configuration Core interacts with both the Audio CODEC on Altera DE2 Board and the Video input on the DE2 Board.
- It provides a convenient way for configuring and Initializing the Audio CODEC and Video in chip.
- The Audio/Video configuration core contains registers that store the configuration and serializer which sends the configuration data via the I2C Bus to the audio and video peripheral.
- Configuration for Audio CODEC:

register num / name	**value**	**notes**
r0 / left line in	9'b0 0001 1111	high gain
r1 / right line in	9'b0 0001 1111	high gain
r2 / left headphone out	9'b0 0111 1001	unity gain
r3 / right headphone out	9'b0 0111 1001	unity gain
r4 / analog audio path	9'b0 0001 0100	mic→ADC, DAC on, no bypass or sidetone
r5 / digital audio path	9'b0 0000 0010	de-emphasis at 32 KHz
r6 / power down control	9'b0 0000 0000	all on
r7 / digital audio format	9'b0 0000 0001	MSB first, left justified, slave mode
r8 / sampling control	9'b0 0001 1010	32 KHz, normal mode, 384Fs oversample
r9 / active control	9'b0 0000 0001	activate

Table 4.1 Audio CODEC Configuration

Phase Locked Loop

- Used for generating clock signal for Audio CODEC, SDRAM, and to overcome loading effect on oscillator.

- Phase Locked Loop (PLL) Megacore Function is used to use two PLL out of 4 PLL present on ALTERA Board.
- Audio_PLL
    - Input Clock : 27 MHz
    - Output Clock => 18.432 MHZ (clock to Audio CODEC)
- NiosMemPLL
    - Input Clock : 50 MHz
    - Output Clock =>

        C0 : 50 MHz (clock to other module)

        C1 : 50 MHz & -3 ns phase shift (clock to SDRAM)

FFT Module

- Used for converting time domain Audio Signal into Frequency domain and store result in 512x16 bits memory block.
- FFT Megacore Function
- Transform Length : 256 points
- Data Input Precision : 16 bits
- Twiddle Precision : 16 points
- FFT Engine Architecture : Quad Output
- I/O Data Flow : Streaming

NIOS II Processor

- NIOS-II/f 32-bit RISC processor with
    - 4kb data cache
    - 4kb instruction cache

- Hardware Multiply and Divide
- Barrel Shifter
- Level 2 JTAG debugger Module
- An integrated floating point multiplier.

- SDRAM for memory requirement
- 50MHz clock
- For communication between the CPU and hardware, the PIO (parallel input output) connected to the NIOS are the following:
  - FFTStart: CPU signals FFT Controller to begin sampling (single line output PIO).
  - FFTDone: CPU gets information that FFT transform is complete (single line input PIO).
  - FFTAddr: Specifies the FFT RAM address to read (9 bits wide).
  - FFTExp: Returns the exponent associated with FFT at the specified address in the RAM (6 bits wide).
  - FFTPower: Returns the power spectrum associated with FFT at the specified address in the RAM (16 bits wide).
  - GreenLED: Processor glow green LED is speaker is verified (1 bit output PIO).
  - RedLED: Processor glow red LED is speaker is invalid (1 bit output PIO).

Name	Description	E...	Clock	Base	End	IRQ
⊞ clk_0	Clock Source					
⊞ cpu_0	Nios II Processor		clk_0	0x0100_0800	0x0100_0fff	
⊞ sdram_0	SDRAM Controller		clk_0	0x0080_0000	0x00ff_ffff	
⊞ lcd_0	Altera Avalon LCD 16207		clk_0	0x0100_1070	0x0100_107f	
⊞ jtag_uart_0	JTAG UART		clk_0	0x0100_1088	0x0100_108f	0
⊞ sysid_0	System ID Peripheral		clk_0	0x0100_1080	0x0100_1087	
⊞ FFTStart	PIO (Parallel I/O)		clk_0	0x0100_1060	0x0100_106f	
⊞ FFTDone	PIO (Parallel I/O)		clk_0	0x0100_1050	0x0100_105f	
⊞ FFTAddr	PIO (Parallel I/O)		clk_0	0x0100_1040	0x0100_104f	
⊞ FFTPower	PIO (Parallel I/O)		clk_0	0x0100_1030	0x0100_103f	
⊞ FFTExp	PIO (Parallel I/O)		clk_0	0x0100_1020	0x0100_102f	
⊞ GreenLED	PIO (Parallel I/O)		clk_0	0x0100_1010	0x0100_101f	
⊞ RedLED	PIO (Parallel I/O)		clk_0	0x0100_1000	0x0100_100f	

Figure 4.2 IP Core in Embedded Processor

4.2 Software Design

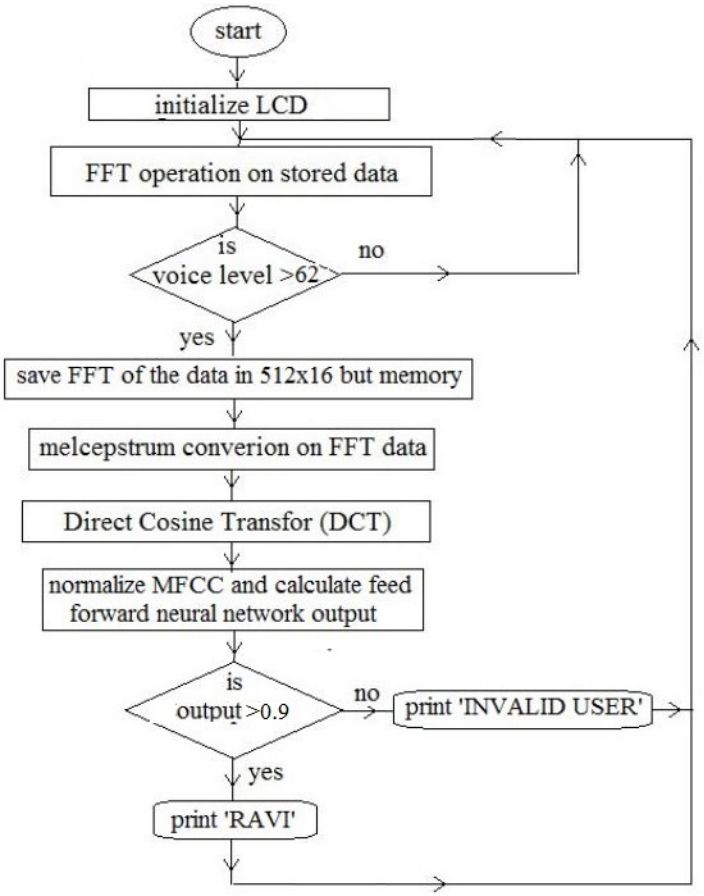


Figure 4.3 Software Flow Diagram

"NIOS II Software Build Tools for Eclipse" Software is used for writing C program for the speech recognition. First LCD is initialized then the code is executing an infinite loop as it's always either expecting the input or processing it. 1024X16 size data (32 msec sampled voice signal from audio CODEC) is continuously saved in the RAM memory. NIOS II processor initiates FFT operation on the saved RAM memory data by FFTStart signal to FFT Module and waits till the completion. To determine whether the input from MIC is silence or someone has spoken, FFT exponent is compared with a value 62 (silence). When the FFT exponent is less than 62, processing continue to next step otherwise processor apply FFT operation on next input voice signal.

The power spectrum of voice signal is saved in 512X16 size memory. The processor starts shifting the stored spectrum using Mel scale and save first 12 frequencies (MFCC) in ROM. Then Direct Cosine Transform (DCT) is applied to get Capstral Coefficient (voice feature) which has also 12 in number.

Next step is Feature Matching. For this purpose two layer Feedforwared Neural Network is used with 12 input nodes (Capstral Coefficient), one hidden layer with 20 nodes, and one output layer. First the Capstral Coefficient is normalized, and Weight and bias for each layer (found during training) is used to get the network output.

Neural Network output is compared with 0.9 value (in training target 1 is assigned for registering the user). If output lies above to 0.9, the person is verified as authenticated, otherwise the system discard the authentication claim.

In the next step, processor control again moves to next input voice signal, and do the same steps.

4.3 Training

Multilayer Feedforward Backpropagation Network has been designed by using Neural Network Tool (nntool). In Matlab, nntool command opens the Network/Data Manager Window, which allows us to import, create, use and export neural networks and data.

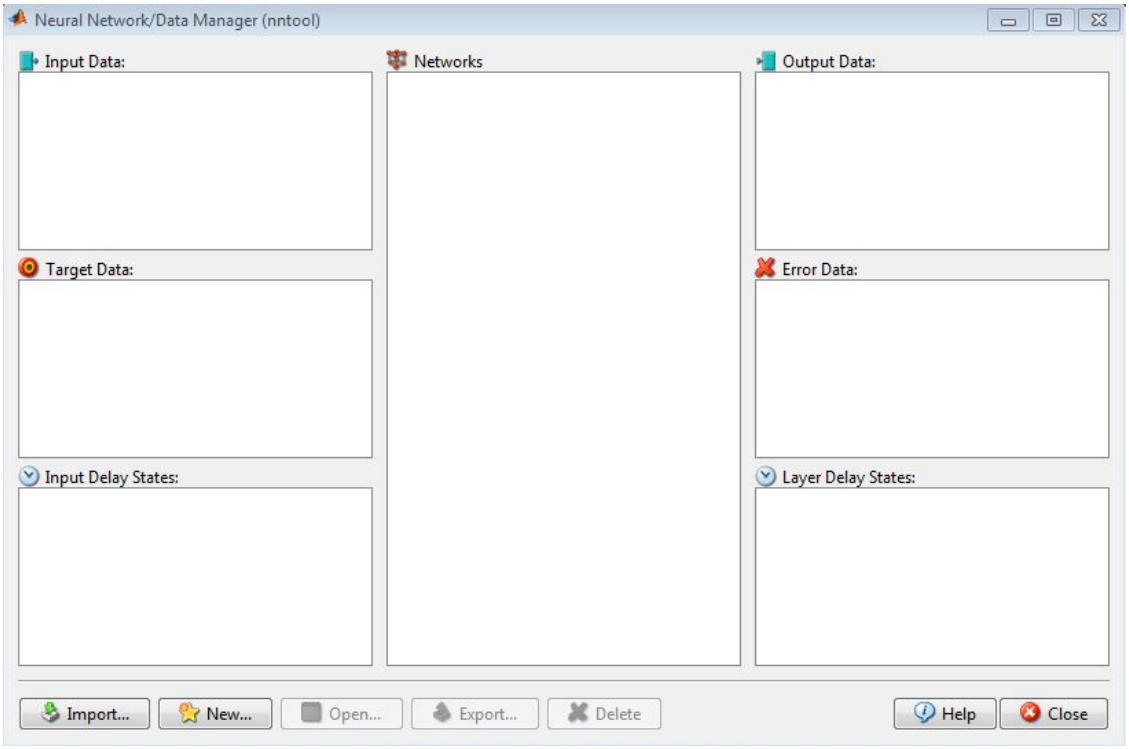


Figure 4.4 nntool Graphical User Interface

For the project, I have created a neural network which has following characteristics:

- One hidden layer with 20 nodes
- tansig sigmoidal activation function
- Levenberg-Marquardt training algorithm

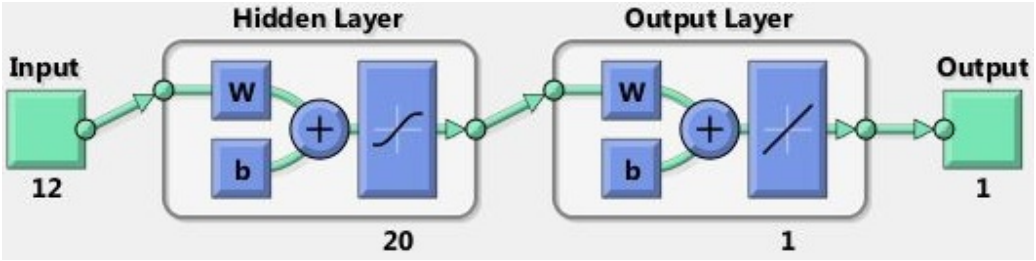


Figure 4.5 Neural Network for the project

Whole training samples has been divided in three subsets: first set is for training (weight adjustment), second set is for learning process control (validation), and third set is for evaluation of the quality of approximation (testing). We have following result while training for the calculation of quality of approximation:

- Mean Square Error (MSE) – it represents difference between network output and target output. Smaller the MSE, better the approximation.
- Pearson's Correlation Coefficient (R) – it represents correlation between network output and target output. Value of R closer to 1, better the approximation.

Chapter 5

Result and Discussion

5.1 Altera DE2 Cyclone II FPGA Resources Utilization

After writing Hardware Description Language in Quartus II and generating NIOS II Processor in Qsys, the whole modules have been compiled. The compilation result shows total FPGA resources used (Figure 5.1), FPGA resources for each individual module (figure 5.2), used IP cores with vendor name (Figure 5.3), and other information. Total 38% Logic Elements, 36% of RAM memory, 40% Multiplier, and 50% PLL are used by the project.

Flow Summary	
Flow Status	Analyzed - Tue May 05 10:05:48 2015
Quartus II 32-bit Version	13.0.1 Build 232 06/12/2013 SP 1 SJ Web Edition
Revision Name	DE2_TOP
Top-level Entity Name	DE2_TOP
Family	Cyclone II
Device	EP2C35F672C6
Timing Models	Final
Total logic elements	12,568 / 33,216 (38 %)
Total combinational functions	10,642 / 33,216 (32 %)
Dedicated logic registers	7,879 / 33,216 (24 %)
Total registers	7947
Total pins	422 / 475 (89 %)
Total virtual pins	0
Total memory bits	173,056 / 483,840 (36 %)
Embedded Multiplier 9-bit elements	28 / 70 (40 %)
Total PLLs	2 / 4 (50 %)

Figure 5.1 Overall resources used in Cyclone II FPGA

Entity	Logic Cells	Dedicated Logic Registers	I/O Registers	Memory Bits	M4Ks	DSP Elements	DSP 9x9	DSP 18x18	Pins	LUT-Only LCs	Register-Only LCs	LUT/Register LCs
Cyclone II: EP2C35F672C6												
DE2_TOP	12568 (1)	7879 (0)	68 (68)	173056	50	28	14	422	0	4689 (1)	1926 (0)	5953 (0)
AudioRAM:audRAM	941 (28)	16 (16)	0 (0)	16384	4	0	0	0	0	925 (12)	0 (0)	16 (16)
ram1024x16:audsamp	0 (0)	0 (0)	0 (0)	16384	4	0	0	0	0	0 (0)	0 (0)	0 (0)
hammingrom:hammrom	913 (913)	0 (0)	0 (0)	0	0	0	0	0	0	913 (913)	0 (0)	0 (0)
hanningrom:hannrom												
sld_hub:auto_hub	197 (1)	109 (0)	0 (0)	0	0	0	0	0	0	88 (1)	18 (0)	91 (0)
VoiceRecognizer:cpu	6931 (0)	3982 (0)	0 (0)	86784	28	4	2	0	0	2949 (0)	767 (0)	3215 (0)
FFTController:fftc	4261 (131)	3624 (51)	0 (0)	69888	18	24	12	0	0	637 (69)	1136 (0)	2488 (78)
FFT2:afft	4147 (6)	3573 (0)	0 (0)	61696	16	24	12	0	0	568 (0)	1136 (0)	2443 (6)
ram512x16:fftpow	0 (0)	0 (0)	0 (0)	8192	2	0	0	0	0	0 (0)	0 (0)	0 (0)
pzdyqx:nabboc	124 (0)	72 (0)	0 (0)	0	0	0	0	0	0	52 (0)	2 (0)	70 (0)
Audio_PLL:p1	0 (0)	0 (0)	0 (0)	0	0	0	0	0	0	0 (0)	0 (0)	0 (0)
NiosMemPLL:p2	0 (0)	0 (0)	0 (0)	0	0	0	0	0	0	0 (0)	0 (0)	0 (0)
I2C_AV_Config:u3	95 (48)	61 (38)	0 (0)	0	0	0	0	0	0	34 (10)	3 (0)	58 (38)
AUDIO_DAC_ADC:u4	18 (18)	15 (15)	0 (0)	0	0	0	0	0	0	3 (3)	0 (0)	15 (15)

Figure 5.2 Resources used by each individual module in Cyclone II FPGA

	Entity	IP Component Name	Version	IP File	Vendor
	ram1024x16	RAM: 2-PORT	13.0	ram1024x16/ram1024x16.qip	Altera
	FFT2	FFT	13.0	FFT/FFT2.qip	Altera
	ram512x16	RAM: 2-PORT	13.0	ram512x16/ram512x16.qip	Altera
	Audio_PLL	ALTPLL	13.0	Audio_PLL/Audio_PLL.qip	Altera
	NiosMemPLL	ALTPLL	13.0	NiosMemPLL/NiosMemPLL.qip	Altera
	N/A	Nios II Processor (6AF7_00A2)	N/A		Altera
	N/A	FFT	N/A		Altera

Figure 5.3 Altera IP core in Cyclone II FPGA

5.2 Training (nntool) result

For training purpose I have taken 100 voice samples from myself and 100 voice sample from my 3 friends. We uttered 'HI' on the Mic. The system generated MFCC (voice feature) for each utterance. Then I copied all MFCC in Matlab. These MFCC is input for the training purpose. Then I created target in which 1 is set for my input data and -1 for my friends input data. Then I opened nntool and set all the parameter for the

training and started training. Figure 5.4 shows training summary: Neural Network Architecture, Algorithm, and progress.

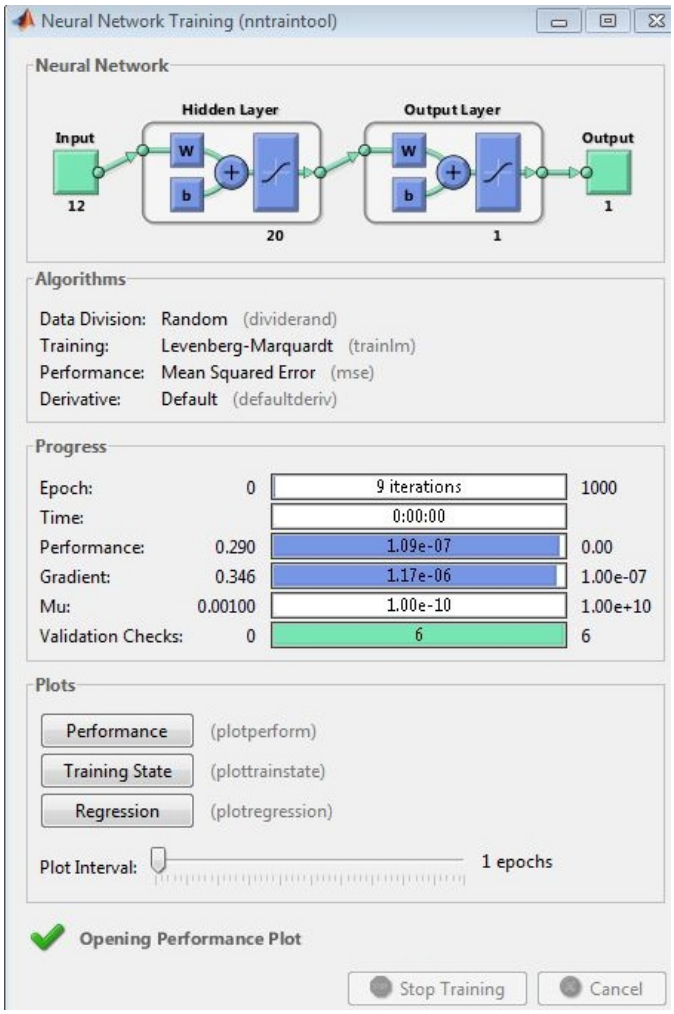


Figure 5.4 nntool Neural Network Summery

Figure 5.5 shows Regression plot which compares training output and predicted output data for group of training, validation, and testing data. Figure 5.6 shows Mean Squared Error (MSE) for the group of training, validation, and testing data. For better neural network, the slope in regression plot should be 1, Y-intercept should be 0, and MSE should be 1. I am getting 'All Regression' value 0.946 (very near to 1) MSE value

very near to 1 at 3 epocs for validation and testing data. So I can conclude that my Neural Network is satisfactory.

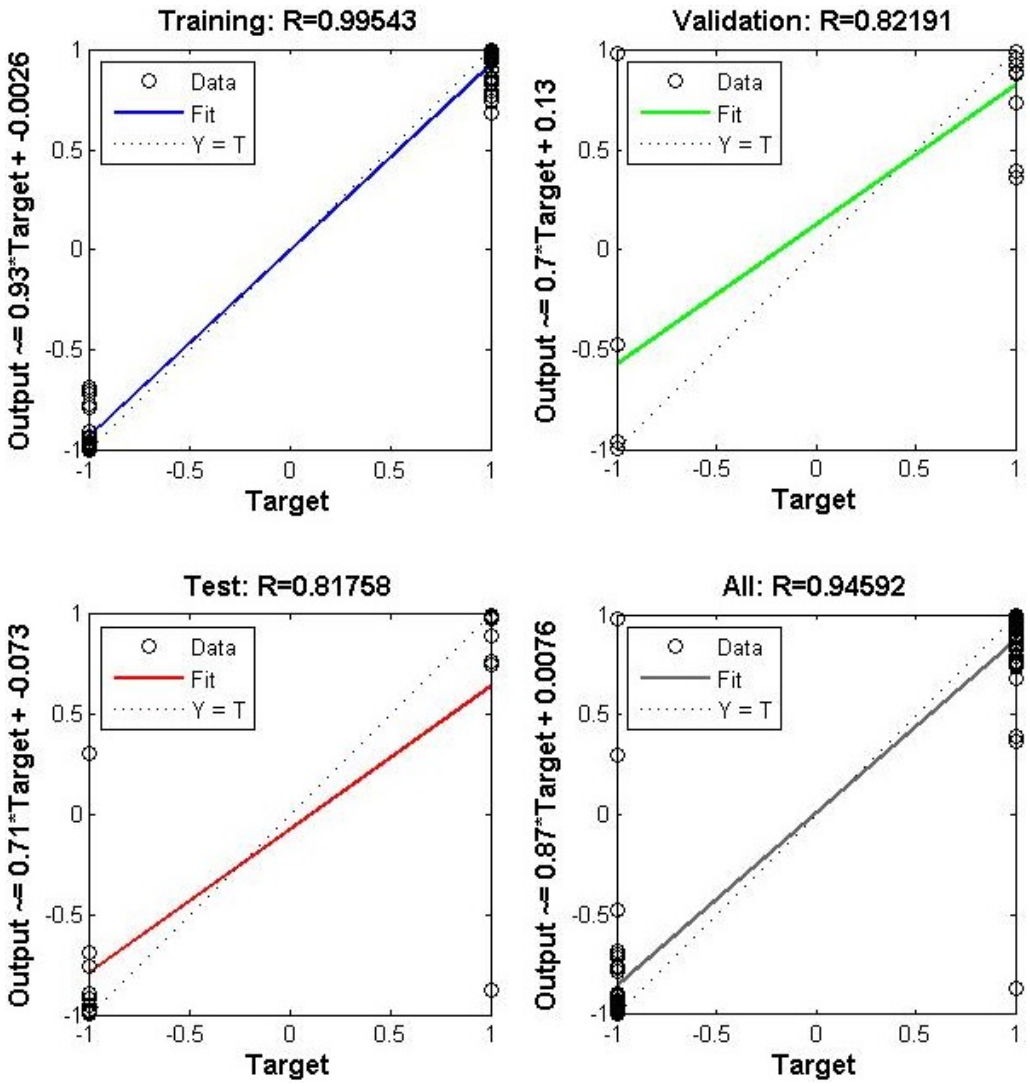


Figure 5.5 Training Network Regression Plot

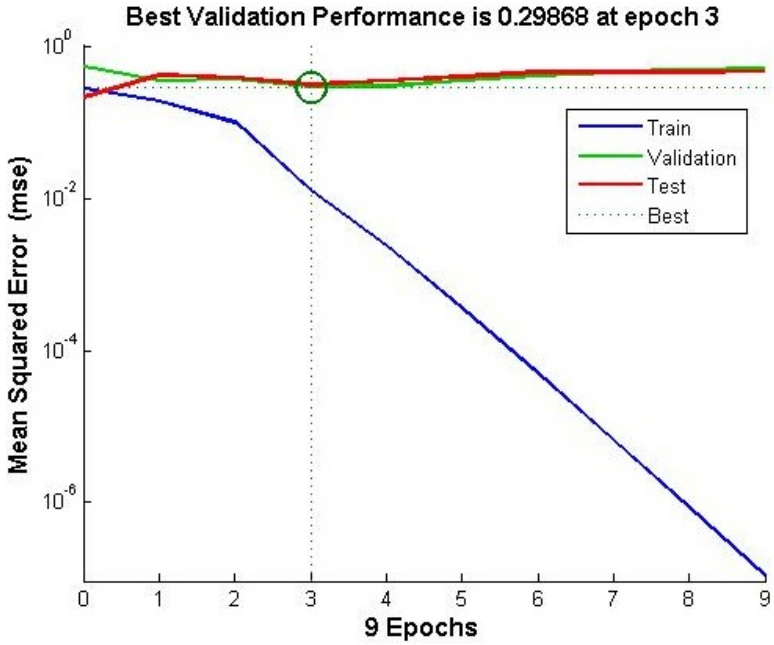


Figure 5.6 Network Training Performances

5.3 Speaker Verificaiton

For testing the designed system I and my friends have uttered 'HI' several times on Mic and the result that I have found is shown in table 6.1.

Speaker	Number of Attempt	RAVI	INVALID USER	Accuracy
RAVI	100	90	10	90%
OTHER	100	12	88	88%

Table 6.1 Accuracy of the designed SOC Audio Processor

Chapter 6

Conclusion and Future Scope

6.1 Conclusion

SoC for an Audio Processing is designed successfully which verify speaker by taking his/her voice on Altera DE2 board. Mel Frequency Capestral Coefficient is used to extract voice feature and Backpropagation Neural Network is used for training and to develop feature matching network. the designed system utilizes 38% Logic Elements, 36% RAM memory, 40% Multiplier, and 50% PLL on Altera DE2 board's FPGA (Cyclone II) resources. Training is done on Matlab by using NNTOOL and it generated neural network which has 0.946 Regression and 1.07×10^{-7} MSE. This neural network is used for Feature Mapping application on the system. The complete system is tested in real time and the system gives 89% accuracy.

6.2 Scope for Future Work

- To implement training unit on SoC
- To improve noise reduction method for better accuracy
- To implement more robust training and recognition method for better accuracy

Bibliography

[1] Saleh, Resve, Steve Wilton, Shahriar Mirabbasi, Alan Hu, Mark Greenstreet, Guy Lemieux, Partha Pratim Pande, Cristian Grecu, and Andre Ivanov. "System-on-chip: reuse and integration." *Proceedings of the IEEE* 94, no. 6 (2006): 1050-1069.

[2] Pollack, Irwin, James M. Pickett, and William H. Sumby. "On the identification of speakers by voice." *the Journal of the Acoustical Society of America* 26, no. 3 (1954): 403-406.

[3] Shearme, J. N., and J. N. Holmes. "An experiment concerning the recognition of voices." *Language and Speech* 2, no. 3 (1959): 123-131.

[4] Peskin, Barbara, Jiri Navratil, Joy Abramson, Douglas Jones, David Klusacek, Douglas A. Reynolds, and Bing Xiang. "Using prosodic and conversational features for high-performance speaker recognition: Report from JHU WS'02." In *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, vol. 4, pp. IV-792. IEEE, 2003.

[5] Yegnanarayana, B., S. Prasanna, Jinu Mariam Zachariah, and Cheedella S. Gupta. "Combining evidence from source, suprasegmental and spectral features for a fixed-text speaker verification system." *Speech and Audio Processing, IEEE Transactions on* 13, no. 4 (2005): 575-582.

[6] Gupta, CHEEDELLA S. "Significance of source features for speaker recognition." *Master's thesis, Indian Institute of Technology Madras, Dept. of Computer Science and Engg., Chennai, India* (2003).

[7] Zhonghua, Fu, and Zhao Rongchun. "An overview of modeling technology of speaker recognition." In *Neural Networks and Signal Processing, 2003.*

Proceedings of the 2003 International Conference on, vol. 2, pp. 887-891. IEEE, 2003.

[8] Saha, Goutam, and U. Yadhunandan. "Modified Mel-Frequency Cepstral Coefficient." In *Proceedings of the IASTED*. 2004.

[9] Picone, Joseph W. "Signal modeling techniques in speech recognition."*Proceedings of the IEEE* 81, no. 9 (1993): 1215-1247.

[10] Childers, Donald G., David P. Skinner, and Robert C. Kemerait. "The cepstrum: A guide to processing." *Proceedings of the IEEE* 65, no. 10 (1977): 1428-1443.

[11] Noll, A. Michael. "Cepstrum pitch determination." *The journal of the acoustical society of America* 41, no. 2 (1967): 293-309.

[12] Kinnunen, Tomi, Evgeny Karpov, and Pasi Franti. "Real-time speaker identification and verification." *Audio, Speech, and Language Processing, IEEE Transactions on* 14, no. 1 (2006): 277-288.

[13] Nedevschi, Sergiu, Rabin K. Patra, and Eric A. Brewer. "Hardware speech recognition for user interfaces in low cost, low power devices." In *Design Automation Conference, 2005. Proceedings. 42nd*, pp. 684-689. IEEE, 2005.

[14] Vu, Ngoc-Vinh, Jim Whittington, Hua Ye, and John Devlin. "Implementation of the MFCC front-end for low-cost speech recognition systems." In *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*, pp. 2334-2337. IEEE, 2010.

[15] Nijhawan, Geeta, and M. K. Soni. "A Comparative Study of Two Different Neural Models For Speaker Recognition Systems." *International Journal of Innovative Technology and Exploring Engineering, ISSN*: 2278-3075, Volume-1, Isseu-1, June 2012.

[16] Bin Hj Salam, M. S., Dzulkifli Mohamad, and Sheikh Hussain Shaikh Salleh. "Temporal Speech Normalization Methods Comparison in Speech Recognition Using Neural Network." In *Soft Computing and Pattern Recognition, 2009. SOCPAR'09. International Conference of*, pp. 442-447. IEEE, 2009.

[17] Seddik, Hassen, Amel Rahmouni, and Mounir Sayadi. "Text independent speaker recognition using the Mel frequency cepstral coefficients and a neural network classifier." In *Control, Communications and Signal Processing, 2004. First International Symposium on*, pp. 631-634. IEEE, 2004.